Threads

CS1812/13: Object Oriented Programming II
Dr Reuben Rowe and Dr Matteo Sammartino
(based on slides by Dr Johannes Kinder)

Concurrency

- Code can execute concurrently
  - Multiple cores allow to execute code at the same time
  - Operating systems can execute code concurrently by time sharing, running and stopping programs in a round-robin fashion
- Examples of concurrency
  - Keep an application's UI responsive while it's calculating
  - Let a server communicate with many clients at the same time
  - Execute expensive computations on multiple cores

Processes

- Processes: an operating system concept
  - Each program runs in its own process
  - Several processes can run in parallel (e.g., on multiple cores)
  - Some programs run in multiple processes (e.g., web server)
- Process isolation
  - Processes cannot access each other's memory
  - Only explicit communication (sockets, pipes, shared memory)

Threads

- Light-weight processes inside a process
  - Each process has at least one thread
  - Several threads can run in parallel
- Separate logical threads of execution
  - E.g., animate buttons while rendering a web page in a browser
- Threads are not isolated and can communicate freely
  - Threads share access to instance and class variables
    - Requires synchronisation